

Did you evaluate your ontology? OOPS! (OntOlogy Pitfall Scanner!)

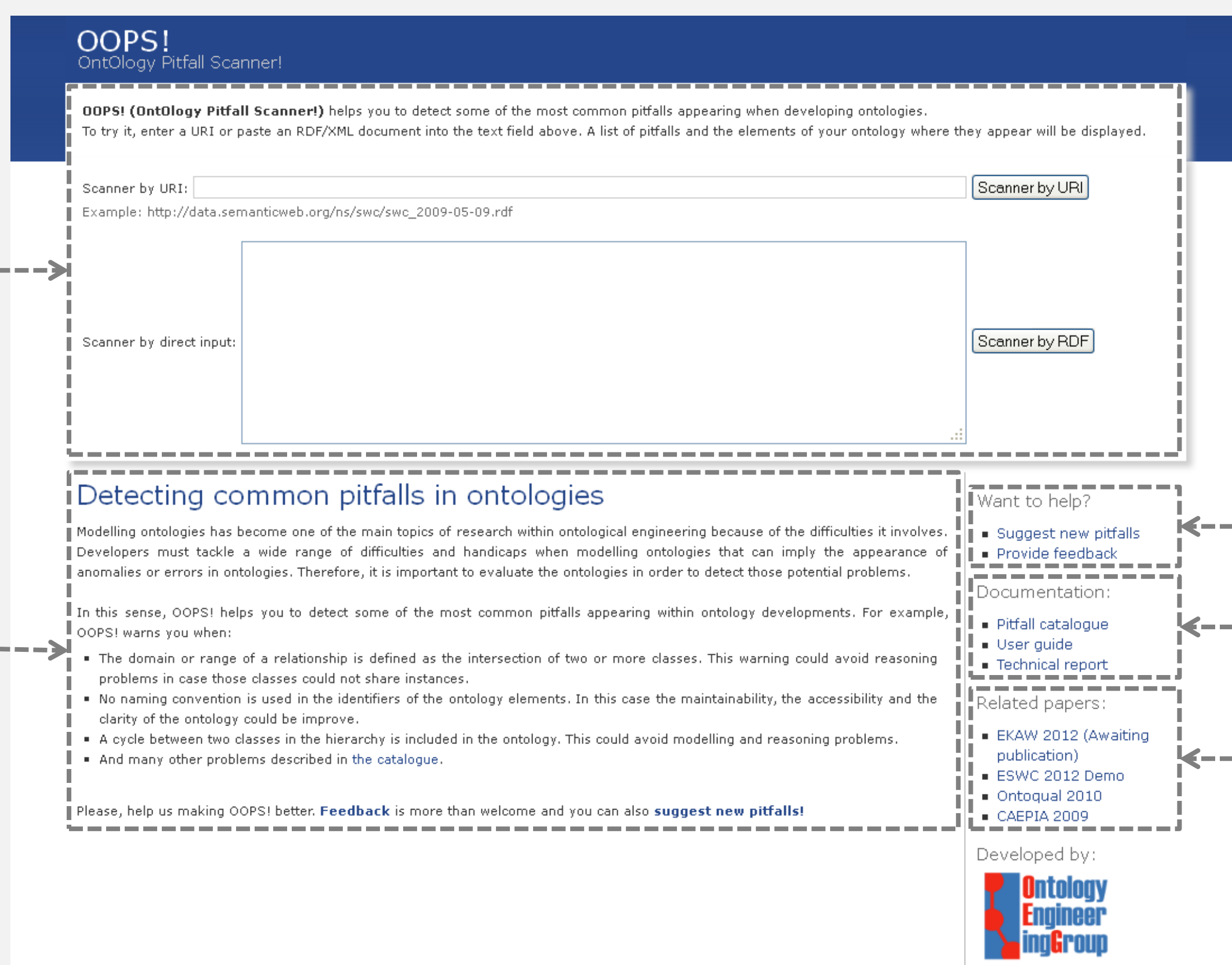
María Poveda-Villalón, Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez
Ontology Engineering Group (OEG). Universidad Politécnica de Madrid
{mpoveda, mcsuarez, asun}@fi.upm.es

Abstract

The application of methodologies for building ontologies can improve ontology quality. However, such quality is not guaranteed because of the difficulties involved in ontology modelling. These difficulties are related to the inclusion of anomalies or bad practices within the ontology development. In this context, our aim is to describe **OOPS! (OntOlogy Pitfall Scanner!)**, a tool for detecting pitfalls in ontologies.

Getting started

<http://www.oeg-upm.net/oops>



OOPS! Ontology Pitfall Scanner!

OOPS! (Ontology Pitfall Scanner) helps you to detect some of the most common pitfalls appearing when developing ontologies. To try it, enter a URI or paste an RDF/XML document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI: Scanner by URI

Example: http://data.semanticweb.org/ns/swc/swc_2009-05-09.rdf

Scanner by direct input: Scanner by RDF

Detecting common pitfalls in ontologies

Modelling ontologies has become one of the main topics of research within ontological engineering because of the difficulties it involves. Developers must tackle a wide range of difficulties and handicaps when modelling ontologies that can imply the appearance of anomalies or errors in ontologies. Therefore, it is important to evaluate the ontologies in order to detect those potential problems.

In this sense, OOPS! helps you to detect some of the most common pitfalls appearing within ontology developments. For example, OOPS! warns you when:

- The domain or range of a relationship is defined as the intersection of two or more classes. This warning could avoid reasoning problems in case those classes could not share instances.
- No naming convention is used in the identifiers of the ontology elements. In this case the maintainability, the accessibility and the clarity of the ontology could be improve.
- A cycle between two classes in the hierarchy is included in the ontology. This could avoid modelling and reasoning problems.
- And many other problems described in the catalogue.

Please, help us making OOPS! better. **Feedback** is more than welcome and you can also **suggest new pitfalls!**

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

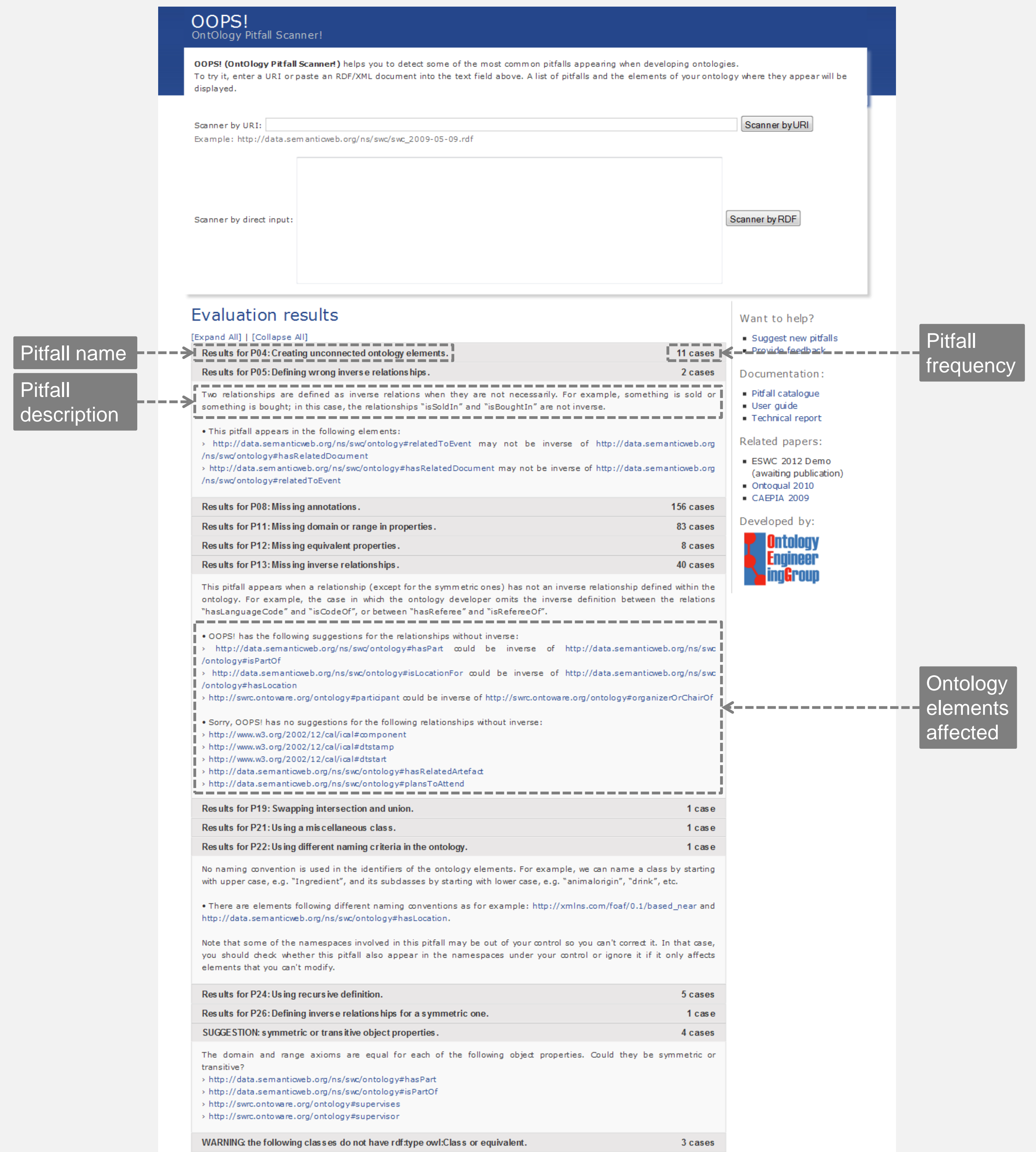
- Pitfall catalogue
- User guide
- Technical report

Related papers:

- EKAW 2012 (Awaiting publication)
- ESWC 2012 Demo
- Ontoqual 2010
- CAEPIA 2009

Developed by: **Ontology Engineering Group**

Scanning ontologies



OOPS! Ontology Pitfall Scanner!

OOPS! (Ontology Pitfall Scanner) helps you to detect some of the most common pitfalls appearing when developing ontologies. To try it, enter a URI or paste an RDF/XML document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI: Scanner by URI

Example: http://data.semanticweb.org/ns/swc/swc_2009-05-09.rdf

Scanner by direct input: Scanner by RDF

Evaluation results

[Expand All] [Collapse All]

Results for P10: Creating unconnected ontology elements. 11 cases

Results for P05: Defining wrong inverse relationships. 2 cases

Two relationships are defined as inverse relations when they are not necessarily. For example, something is sold or something is bought; in this case, the relationships "isSoldIn" and "isBoughtIn" are not inverse.

This pitfall appears in the following elements:

- <http://data.semanticweb.org/ns/swc/ontology#relatedToEvent> may not be inverse of <http://data.semanticweb.org/ns/swc/ontology#hasRelatedDocument>
- <http://data.semanticweb.org/ns/swc/ontology#hasRelatedDocument> may not be inverse of <http://data.semanticweb.org/ns/swc/ontology#relatedToEvent>

Results for P08: Missing annotations. 156 cases

Results for P11: Missing domain or range in properties. 83 cases

Results for P12: Missing equivalent properties. 8 cases

Results for P13: Missing inverse relationships. 40 cases

This pitfall appears when a relationship (except for the symmetric ones) has not an inverse relationship defined within the ontology. For example, the case in which the ontology developer omits the inverse definition between the relations "hasLanguageCode" and "isCodeOf", or between "hasRefers" and "isRefersOf".

OOPS! has the following suggestions for the relationships without inverse:

- <http://data.semanticweb.org/ns/swc/ontology#hasPart> could be inverse of <http://data.semanticweb.org/ns/swc/ontology#hasLocation>
- <http://swc.ontoware.org/ontology#participant> could be inverse of <http://swc.ontoware.org/ontology#organizerOrChairOf>

Sorry, OOPS! has no suggestions for the following relationships without inverse:

- <http://www.w3.org/2002/12/cal/ical#component>
- <http://www.w3.org/2002/12/cal/ical#timestamp>
- <http://www.w3.org/2002/12/cal/ical#dataref>
- <http://data.semanticweb.org/ns/swc/ontology#hasRelatedDateRef>
- <http://data.semanticweb.org/ns/swc/ontology#plansToAttend>

Results for P19: Swapping intersection and union. 1 case

Results for P21: Using a miscellaneous class. 1 case

Results for P22: Using different naming criteria in the ontology. 1 case

No naming convention is used in the identifiers of the ontology elements. For example, we can name a class by starting with upper case, e.g. "Ingredient", and its subclasses by starting with lower case, e.g. "animalorigin", "drink", etc.

There are elements following different naming conventions as for example: http://xmlns.com/foaf/0.1/based_near and <http://data.semanticweb.org/ns/swc/ontology#hasLocation>.

Note that some of the namespaces involved in this pitfall may be out of your control so you can't correct it. In that case, you should check whether this pitfall also appear in the namespaces under your control or ignore it if it only affects elements that you can't modify.

Results for P24: Using recursive definition. 5 cases

Results for P26: Defining inverse relationships for a symmetric one. 1 case

SUGGESTION: symmetric or transitive object properties. 4 cases

The domain and range axioms are equal for each of the following object properties. Could they be symmetric or transitive?

- <http://data.semanticweb.org/ns/swc/ontology#hasPart>
- <http://data.semanticweb.org/ns/swc/ontology#isPartOf>
- <http://swc.ontoware.org/ontology#supervises>
- <http://swc.ontoware.org/ontology#supervisor>

WARNING: the following classes do not have rdf:type owl:Class or equivalent. 3 cases

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

Related papers:

- ESWC 2012 Demo (awaiting publication)
- Ontoqual 2010
- CAEPIA 2009

Developed by: **Ontology Engineering Group**

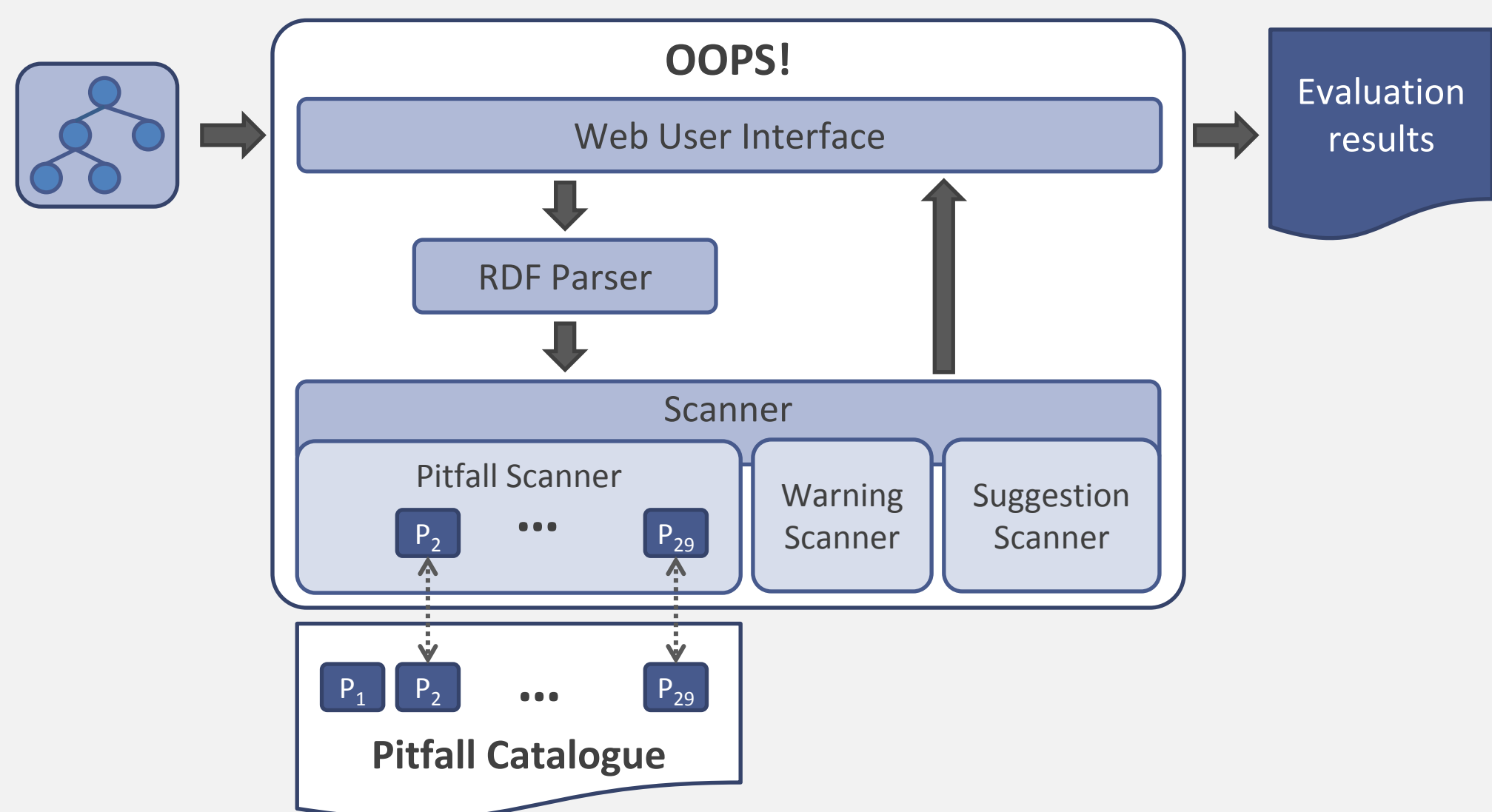
Pitfall name

Pitfall description

Pitfall frequency

Ontology elements affected

Architecture



Pitfall catalogue

- P1. Creating polysemous elements
- P2. Creating synonyms as classes
- P3. Creating the relationship "is" instead of using "rdfs:subClassOf", "rdf:type" or "owl:sameAs"
- P4. Creating unconnected ontology elements
- P5. Defining wrong inverse relationships
- P6. Including cycles in the hierarchy
- P7. Merging different concepts in the same class
- P8. Missing annotations
- P9. Missing basic information
- P10. Missing disjointness
- P11. Missing domain or range in properties
- P12. Missing equivalent properties
- P13. Missing inverse relationships
- P14. Misusing "owl:allValuesFrom"
- P15. Misusing "not some" and "some not"
- P16. Misusing primitive and defined classes
- P17. Specializing too much a hierarchy
- P18. Specifying too much the domain or the range
- P19. Swapping intersection and union
- P20. Misusing ontology annotations
- P21. Using a miscellaneous class
- P22. Using different naming criteria in the ontology
- P23. Using incorrectly ontology elements
- P24. Using recursive definition
- P25. Defining a relationship inverse to itself
- P26. Defining inverse relationships for a symmetric one
- P27. Defining wrong equivalent relationships
- P28. Defining wrong symmetric relationships
- P29. Defining wrong transitive relationships

Conclusions

- OOPS! represents a step forward within ontology evaluation tools as :
 - a) it **enlarges the list of errors detected** by most recent and available works
 - b) it is fully **independent** of any ontology **development environment**
 - c) it works with **main web browsers** (Firefox, Chrome, Safari and IE)
- It is being **tested by OEG** members and **used by** ontology developers who belong to **different organizations** (such as AtoS, Tecnalia, Departament Arquitectura, La Salle at Universitat Ramon Llull, etc.).
- It is **freely available** to users on the Web. Everyone can test it, provide **feedback**, **suggest new pitfalls** to be included in the catalogue and implemented into the tool.

Next steps

- To enlarge the pitfall catalogue by including:
 - new errors that might be **detected** during our research
 - pitfalls **suggested by users**
- To **group** and **classify** pitfalls by categories and allow the user to **choose** categories or particular pitfalls to check the ontology against
 - OOPS! more flexible and adaptable to specific user needs
- To increase OOPS! features with **guidelines** about how to **solve** each **pitfall**
 - To facilitate the task of repairing the ontology after the diagnosis
- To associate **priority levels** to each pitfall according to their different types of consequences they can convey
 - To prioritize actions to be taken during the repairing task
- To make **REST services** available
 - To allow other developments to use and integrate the pitfall scanner functionalities within their applications
- To allow users to **define pitfalls** following a **formal language**, according with their particular quality criteria
 - To allow users to execute OOPS! in a customized way

Acknowledgments

This work has been partially supported by the Spanish projects BabelData (TIN2010-17550) and BuscaMedia (CENIT 2009-1026).